

QualiPSo MOSST Operating Manual

March 2012

Information-technology Promotion Agency, Japan

About this document

This document describes "MOSST", one of the tools for evaluating reliability trustworthiness of OSS, as well as the basic operation procedure up to showing the evaluation result of OSS.

Intended audience of this document

This document is intended for IT system professionals who are involved in use of OSS. The description in this document assumes that the reader has a basic knowledge of software development and installation.

About trademarks and registered trademarks

The names of trades and products used in this document may be the trademarks or registered trademarks of respective owners.

Furthermore, TM and (R) are not mentioned in each case in this document.

Change History

Date	Description
2011.8.24	Initial version
2012.3.30	English translation

Limitations

1. About StatCVS

As of August 2011, StatCVS does not support project which uses Japanese JIS code set for commit log.

StatCVS accesses commit log of OSS project to acquire statistical values, which does not support "0x1B" included. An example of character code set which is subject to restrictions of this byte string "0x1B" is JIS code set.

2. About confirmed operating environment

Operation has been confirmed only under local environment within the server.

Table of Contents

1.	Introduction.....	1
1.1.	Contents of this document.....	1
1.2.	Intended audience of this document	1
1.3.	About MOSST	1
1.4.	About use of MOSST	2
1.5.	MOSST architecture.....	2
2.	MOSST Overview.....	3
2.1.	Measurement indicators supported for evaluation using MOSST	3
2.2.	Operation overview	4
2.2.1.	Common operations for each tool	4
2.2.2.	Operation for JaBUTi	4
2.3.	About OSS available for evaluation using MOSST tool.....	6
2.3.1.	Registered information and evaluation tool.....	6
3.	Access to each MOSST tool.....	7
4.	Basic operations of MOSST	8
4.1.	Registering project	8
4.2.	Executing evaluation	14
4.2.1.	Extracting project information	14
4.2.2.	Preparing automated test (for JaBUTi only)	17
4.2.3.	Automated test (for JaBUTi only)	17
4.2.4.	Analyzing project information.....	23
4.3.	Verifying evaluation result	26
5.	Details of evaluation result.....	29
6.	Changing project	32
7.	Changing KPI definition	34
7.1.	Changing KPI threshold value.....	34
8.	Frequently asked questions about troubles (FAQ).....	39
8.1.	When executing Extractor on Spaga4Q	39
8.2.	When showing report of evaluation result	43
8.3.	Others	44
9.	References	45

1. Introduction

1.1. Contents of this document

OSS (Open Source Software) has gained wide application, as fundamentals of IT systems, and its trustworthiness and sustainability gains significant importance. Furthermore, evaluation from various perspectives should be required not only for quality of software itself, but also for maintainability, potential legal issues, and support scheme of development community.

IPA participates in QualiPSo (Quality Platform for Open Source Software) network initiated by European Commission (EC), and facilitates tools for evaluating OSS trustworthiness under its international cooperation.

This document describes basic operations of the tools named "MOSST", for its use in OSS evaluation. For other function details which are not covered by this document, please refer to relevant documents such as user manual of each tool.

1.2. Intended audience of this document

This document is intended for IT system professionals who are involved in use of OSS. The description in this document assumes that the reader has a basic knowledge of software development and evaluation.

1.3. About MOSST

MOSST (Model for OSS trustworthiness) is an open source software trustworthiness model designed for evaluating quality of OSS products in a quantitative manner. MOSST provides various analyses for source code, as well as user interface with which analysis result can be visually presented. MOSST involves the following tools.

Tool Name	Description
Spago4Q	Evaluation result visualization tool
Quality Platform	Web interface for evaluation tools
Kalibro	C++ code metrics measurement tool
StatCVS/StatSVN	Repository statistics measurement tool
Macxim	Java code metrics measurement tool
JaBUTi	Test coverage measurement tool

For details of each tool used by MOSST, please refer to the following QualiPSo website:

<<http://www.qualipso.org/mosst-champion>>

1.4. About use of MOSST

Tools for use of MOSST are available for download at the following URL:

<<http://www.qualipso.org/mosst-champion>>

For details on how to install them, please refer to "QualiPSo MOSST Installation Manual".

1.5. MOSST architecture

Among several tools which MOSST involves, "Quality Platform" and "Spaga4Q" are major operation interface used in basic steps up to evaluation result display.

- Quality Platform

Quality Platform displays the result obtained from each measurement tool in a comprehensive manner. It has project registration feature intended for use by system administrator.

- Spaga4Q

Spaga4Q is a BI tool mainly intended for system administrator. It is used to extract/analyze project information and performs other tasks.

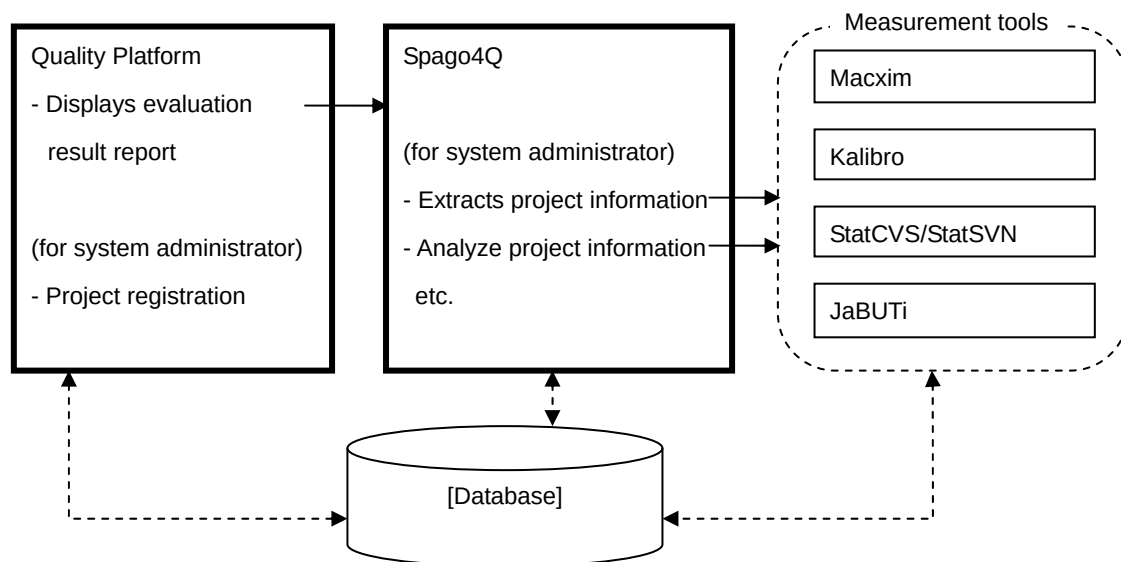
(actual processing is performed by each measurement tool, and results are stored to database).

- Measurement tools

Measurement is performed based on registered project information and configured values.

- Database

Database stores project information and measurement results produced by each tool.



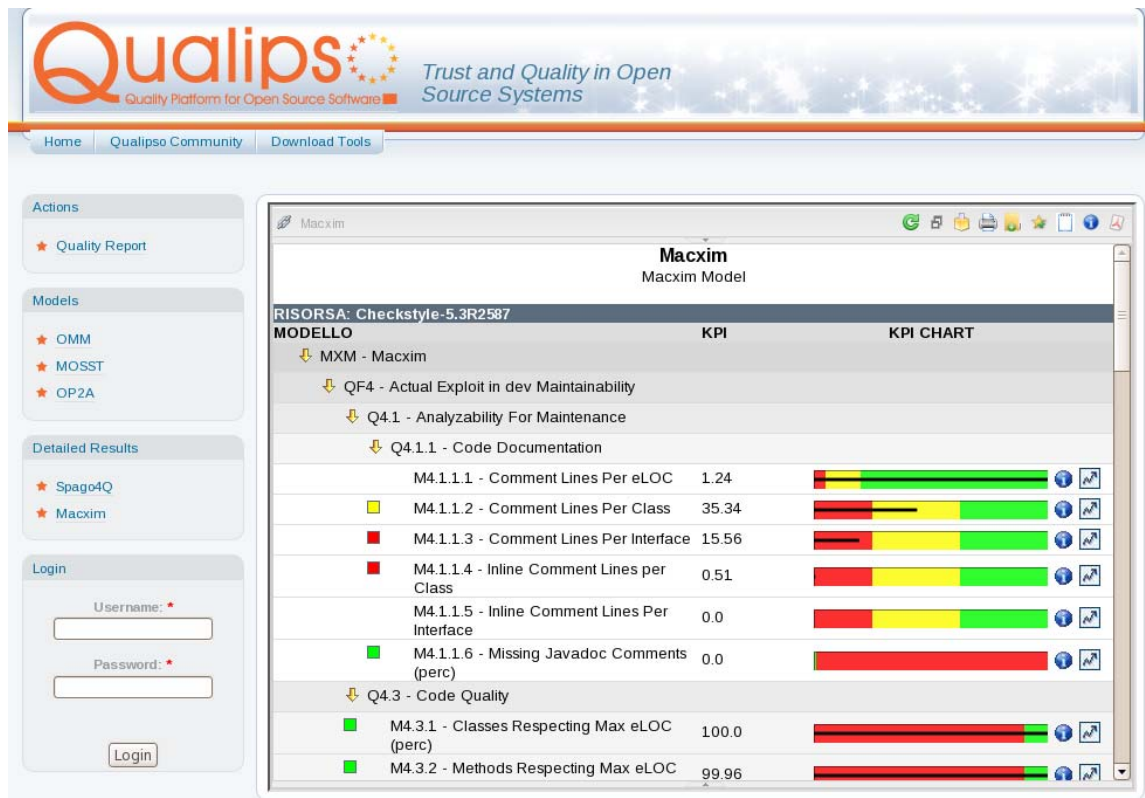
2. MOSST Overview

2.1. Measurement indicators supported for evaluation using MOSST

Measurement indicators available for evaluation using MOSST are as follows:

- C++ code metrics (Kalibro)
- Repository statistics (StatCVS/StatSVN)
- Java code metrics (Macxim)
- Test coverage (JaBUTi)

< Java code metrics (Macxim) evaluation example>



This section provides an overview of operations.

In addition to common operations, only JaBUTi requires its particular step.

OSS evaluation is performed in the following basic workflow:

1. and 2. are operations performed by system administrator. Part of the operations require login as system administrator.

2.2.2. Operation for JaBUTi

1. Register project (system administrator)
2. Execute evaluation (system administrator)
 - 2-1. Extract project
 - 2-1-1. Prepare for automated test
 - 2-1-2. Execute automated test
 - 2-2. Analyze project
3. Verify evaluation result (user)



(This page is blank)

2.3. About OSS available for evaluation using MOSST tool

2.3.1. Registered information and evaluation tool

As described in the previous section, MOSST tool requires registration of OSS project to be evaluated.

Evaluation tool can be selected at registration, while different tools are available depending on OSS programming language/repository type.

Please confirm tools available for evaluation with the following table:

		Programming language			
		C/C++		Java	
Repository type	SVN	StatSVN/CVS Kalibro		StatSVN/CVS JaBUTi Macxim	
	CVS	StatSVN/CVS		StatSVN/CVS JaBUTi	

Example 1) For OSS using programming language "C/C++" and repository type "CVS":

⇒ Tool available for evaluation is "StatSVN/CVS".

Example 2) For OSS using programming language "Java" and repository type "SVN":

⇒ Tools available for evaluation are "StatSVN/CVS" , "JaBUTi" , and "Macxim".

3. Access to each MOSST tool

Examples of startup URL/startup command are shown below.

These examples are access information applicable to installation made in accordance with "QualiPSo MOSST Installation Manual".

Example startup URL

Tool	URL
Spago4Q	http://[Host name]:[Port number]/SpagoBI/
Quality Platform	http://[Host name]:[Port number]/quality-platform/
StatCVS/StatSVN	http://[Host name]:[Port number]/StatToolsService/cacheManager
Macxim	http://[Host name]:[Port number]/macxim

Example startup command

Tool	Command (executed in folder under which jar file exists)
Kalibro	java -jar KalibroDesktop.jar
JaBUTi	java -jar JaBUTi-service-gui-scripts-1.0.jar or sh ./run.sh

4. Basic operations of MOSST

Sections "4.1 Registering project" and "4.2 Executing evaluation" involve operations performed by system administrator.

For operation performed by user, please refer to section "4.3 Verifying evaluation result".

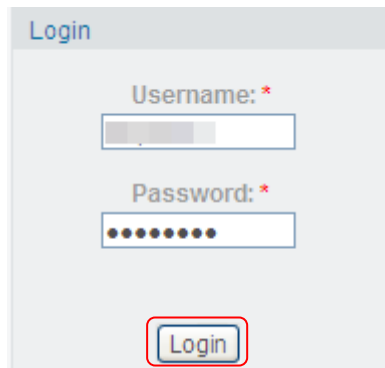
4.1. Registering project

From Quality Platform, register OSS to evaluate.

1. Start and log on to Quality Platform.

At Login section on lower-left of the top page, enter Username/Password (*) and click on [Login].

* Please refer to the section of Quality Platform in "QualiPSo MOSST Installation Manual".

A screenshot of a login form titled "Login". It contains two input fields: "Username: *" and "Password: *". The password field is masked with dots. Below the fields is a "Login" button, which is highlighted with a red rectangle.

2. When login is successful, "Insert Project" appears on Actions menu.

Click on "Insert Project".



3. On "Insert Project" screen, enter required data such as repository URL.

Once required entries are made and [Insert] is clicked, project information is stored to database.

The following part describes major items for repository types of svn and cvs, respectively.

(1) Repository type is svn:

The screenshot shows the 'Insert Project' form in the Qualips application. The form is divided into several sections: Project Information, License, Repository Information, Model Information, Tools Information, Repository URL, Username/Password, Revision number, Start/End dates, Other Informations, and Build Type. Annotations <1> through <8> point to specific fields: <1> points to the Project Name field (containing 'Checkstyle'), <2> points to the Version field (containing '5.3'), <3> points to the Language dropdown (set to 'java'), <4> points to the Type of repository dropdown (set to 'svn'), <5> points to the Model and Tools checkboxes (Macxim, MOSST, OMM, OSLC, JaBUTi, CVSAnalY, StatSVN/CVS, Bicho, JUnit, Kalibro), <6> points to the Repository URL field (containing 'https://checkstyle.svn.sourceforge.net/svnroot/checkstyle/trunk/checkstyle/src/cli'), <7> points to the Username and Password fields, and <8> points to the Revision number field (containing '2587'). The 'Insert' button is highlighted with a red box at the bottom left.

Qualips
Trust and Quality in Open Source Systems

Home | Qualips Community | Download Tools

Actions

- Quality Report
- Insert Project

Models

- OMM
- MOSST
- OP2A

Detailed Results

- Spago4Q
- Macxim

Login

Logged in as: sqadmin

Logout

Insert Project

Project Information

Project Name: * <1>

Version: * <2>

Language: <3>
The programming language.

License:

<4>
From www.opensource.org.

Repository Information

Type of repository: * <4>

Model Information

Model: *

- ☒ MOSST
- ☐ OMM

Tools Information

Tools:

- ☒ Macxim
- ☐ OSLC
- ☐ JaBUTi
- ☐ CVSAnalY
- ☐ StatSVN/CVS
- ☐ Bicho
- ☐ JUnit
- ☐ Kalibro

Repository URL: * <6>

Username: Password: <7>

Revision number: * <8>

Start: End:

Other Informations

Minor Release Pattern:

Major Release Pattern:

CVS Connection Protocol:

Build Type:

[Entry details]

<1> OSS project name

<2> OSS project version

<3> Programming language

Select java or c/c++.

<4> Repository type

Select svn.

<5> Tool selection

- Models (on the left): put a check on MOSST

- Tools (on the right): put a check on evaluation tool

Selected tool will be subject to execution of evaluation.

Select Macxim, JaBUTi, StatSVN/CVS or Kalibro.

Some tools may not be available for selection depending on selection made in <3> and <4>.

<6> URL of OSS project repository

When evaluation is executed, source code at this URL is checked out.

<7> Repository user/password

Enter as required.

<8> Revision eligible for checkout

Enter existing revision.

(2) Repository type is cvs:

Qualips Trust and Quality in Open Source Systems
Quality Platform for Open Source Software

Home | Qualips Community | Download Tools

Actions

- ★ Quality Report
- ★ Insert Project

Models

- ★ OMM
- ★ MOSST
- ★ OP2A

Detailed Results

- ★ Spago4Q
- ★ Macxim

Login

Logged in as: s4qadmin

Logout

Insert Project

Project Information

Project Name: * **<1>**

Version: * **<2>**

Language: **<3>**
The programming language.

License:

<4>
From www.opensource.org.

Repository Information

Type of repository: * **<4>**

Model Information

Model: *

- ☒ MOSST
- ☐ OMM

Tools Information

Tools:

- ☒ Macxim
- ☐ StatSVN/CVS
- ☐ OSLC
- ☐ Bicho
- ☐ JaBUTi
- ☐ JUnit
- ☐ CVSAnalY
- ☐ Kalibro

<5>

Repository URL: *

<6>

Username: **<7>**

Password: **<7>**

Revision number: *

<8>

Start: **<8>**

End: **<8>**

Other Informations

Minor Release Pattern:

Major Release Pattern:

CVS Connection Protocol:

<9>

Build Type:

Insert

[Entry details]

<1> OSS project name

Enter folder name which exists in repository.

In StatCVS, a folder which has this project name is accessed to check out project. Project cannot be checked out if there is no folder with its project name. Please ensure to specify project name which has been registered to CVS repository.

<2> OSS project version

<3> Programming language

Select java or c/c++.

<4> Repository type

Select cvs.

<5> Tool selection

- Models (on the left): put a check on MOSST

- Tools (on the right): put a check on evaluation tool

Selected tool will be subject to execution of evaluation.

Select Macxim, JaBUTi, StatSVN/CVS or Kalibro.

Some tools may not be available for selection depending on selection made in <3> and <4>.

<6> URL of OSS project repository

When evaluation is executed, source code at this URL is checked out.

<7> Repository user/password

Enter as required.

<8> Revision eligible for checkout

Enter existing revision

<9> CVS connection protocol

Select pserver, extssh, pserverssh2 or ext.

4. Verify registered information from Spago4Q.

Log on to Spago4Q as administrator (*), and select KPI Models > Resources Definition to show list.

Verify registered project.

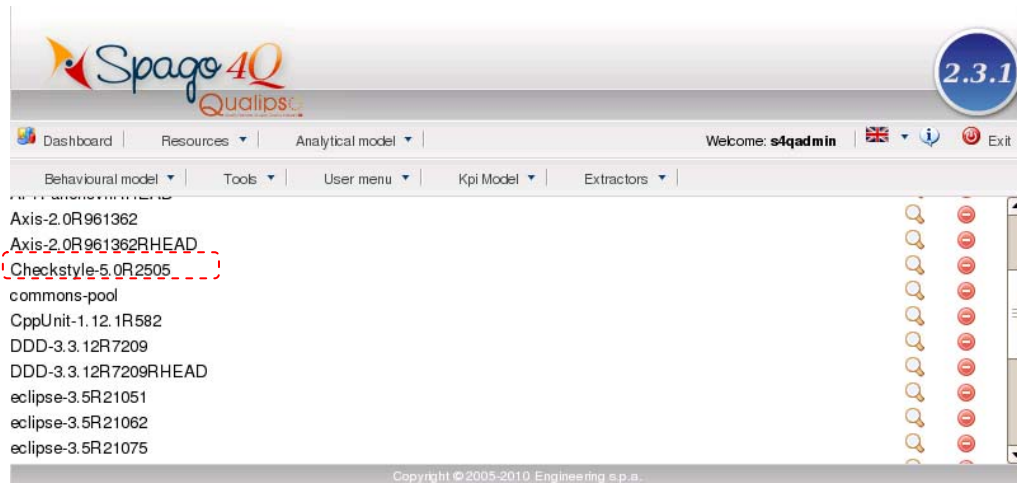
* Please also refer to the section of Spago4Q of "QualiPSO MOSST Installation Manual".

- Example) If the following project is registered, resource name is "Checkstyle-5.0R2505".

Project name: Checkstyle

Version: 5.0

Revision: 2505



4.2. Executing evaluation

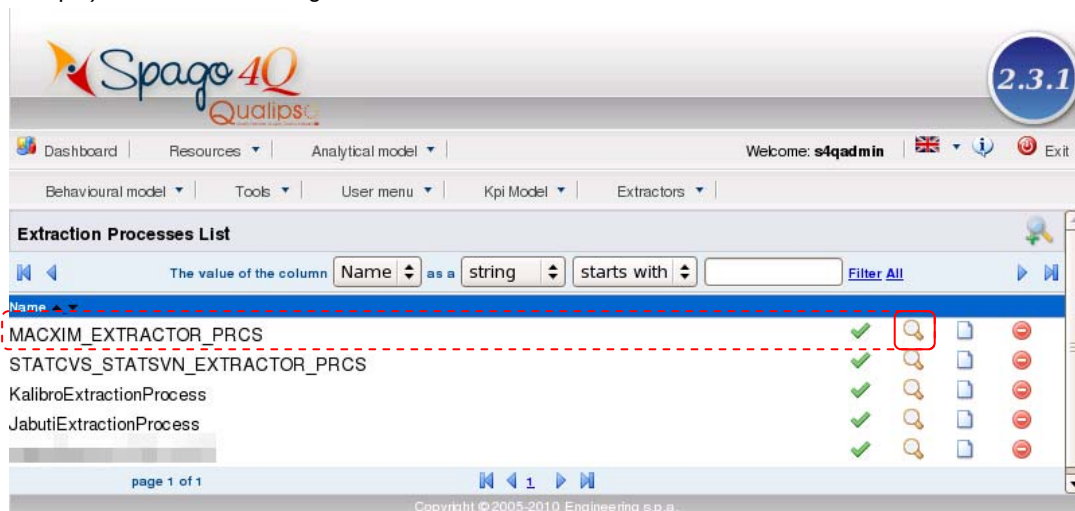
Select evaluation tool and extract/analyze project information. Extract/analyze operations are performed by Extractor processing of Spago4Q tool.

In addition, only JaBUTi requires "automated test" operation to take place between extraction and analysis.

4.2.1. Extracting project information

1. Log on to Spago4Q as administrator.
2. Select Spago4Q > Extractors > Extraction Processes and show list of evaluation tools.
3. Click on Select button  of tool you wish to use for evaluation.

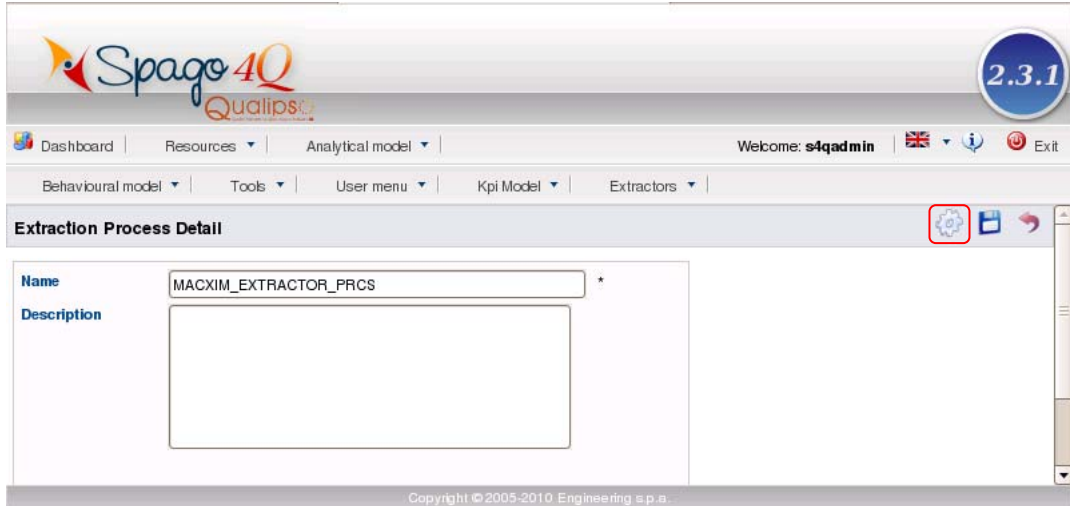
Example) For evaluation using Macxim



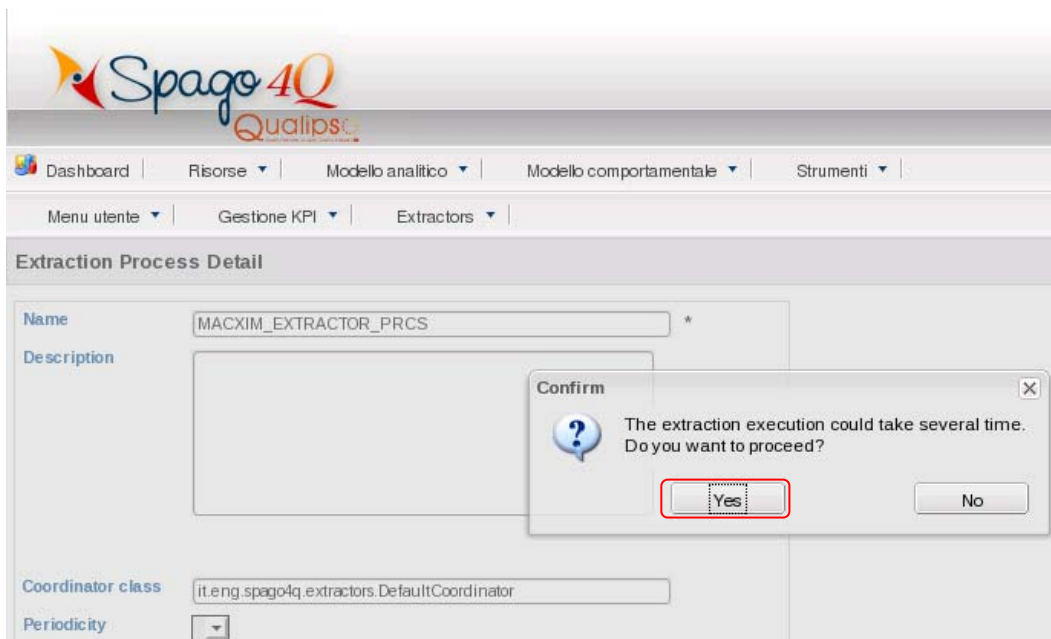
For Name values which correspond to each tool, please refer to the following table:

Evaluation tool	Name
Macxim	MACXIM_EXTRACTOR_PRCs
StatCVS/StatSVN	STATCVS_STATSVN_EXTRACTOR_PRCs
Kalibro	KalibroExtractionProcess
JaBUTi	JaBUTiExtractionProcess

4. Extraction Process Detail screen appears. Click on Save and Execute button .
Example) For evaluation using Macxim

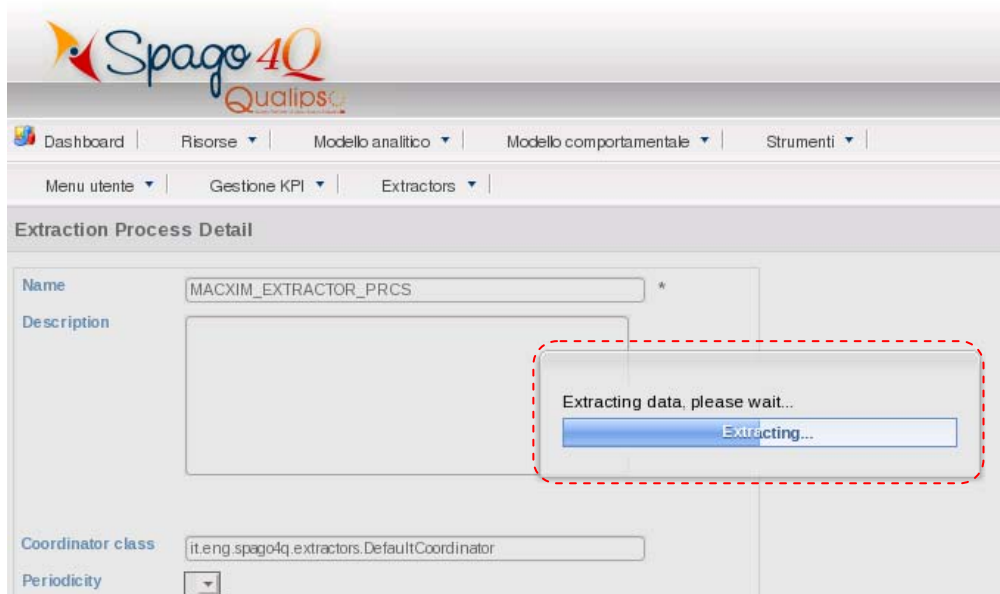


5. Once prompted that "processing may take time", confirm the message and click on "Yes".



6. Progress meter appears on the screen.

In some cases, internal processing may not be finished when progress meter completes.



Processing status can be confirmed on log file.

Example log file) [tomcat_root]/logs/catalina.out
[tomcat_root]/logs/SpagoBI.log

This operation provides access to repository URL registered in section "4.1 Registering project" to check out source code.

* Only for JaBUTi, checkout uses OS command operation. For details, please refer to the next section and beyond.

4.2.2. Preparing automated test (for JaBUTi only)

JaBUTi tool allows to measure test coverage of Java programs. Project environment for executing automated test (build environment for project and test scripts) is required to be prepared and built by the user prior to test execution.

For details on how to prepare and build environment, and on how to run test scripts, please refer to information provided by each OSS project, respectively.

Precautions for the 2nd or later "Execute" processing

While test coverage measurement is being executed (during Execute processing), JaBUTi tool embeds its proprietary processing, which is designed for coverage measurement, onto the class files (*.class) being tested.

Once it is completed, and if measurement is executed once again, the above class files will produce inconsistency and the result of test coverage measurement will become inaccurate.

In order to avoid this problem, at every measurement of test coverage (Execute processing), please clean the build environment (by removing all assets which were generated by the previous build), and rebuild the class files.

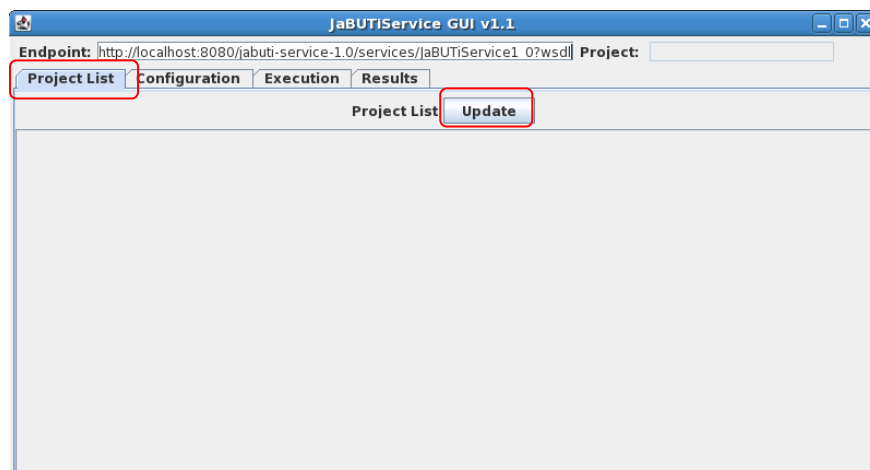
4.2.3. Automated test (for JaBUTi only)

* It may take several hours to complete this process.

1. Start JaBUTi Service GUI.

```
# sh ./run.sh  
or  
# java -jar /jabuti-service-gui-scripts-1.0.jar
```

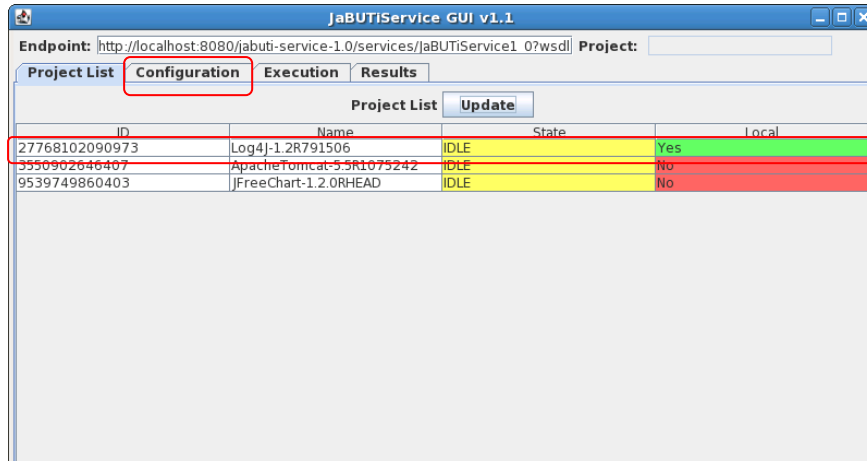
2. Select Project List tab and click on Update button.



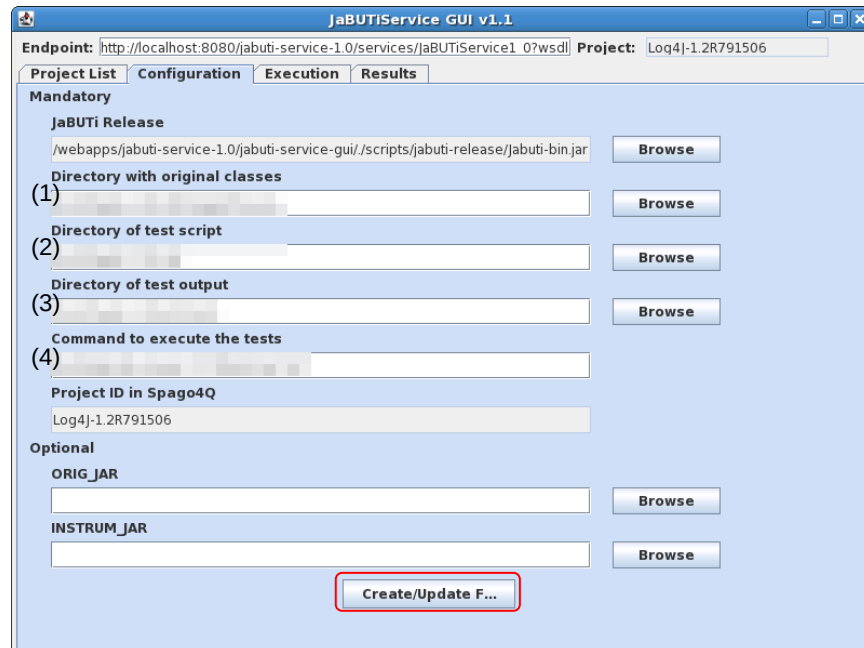
(This page is blank)

3. The list of registered projects appears.

Select the project you wish to evaluate, and select Configuration tab.



4. Once required entries are made, click on [Create/Update F...] button.

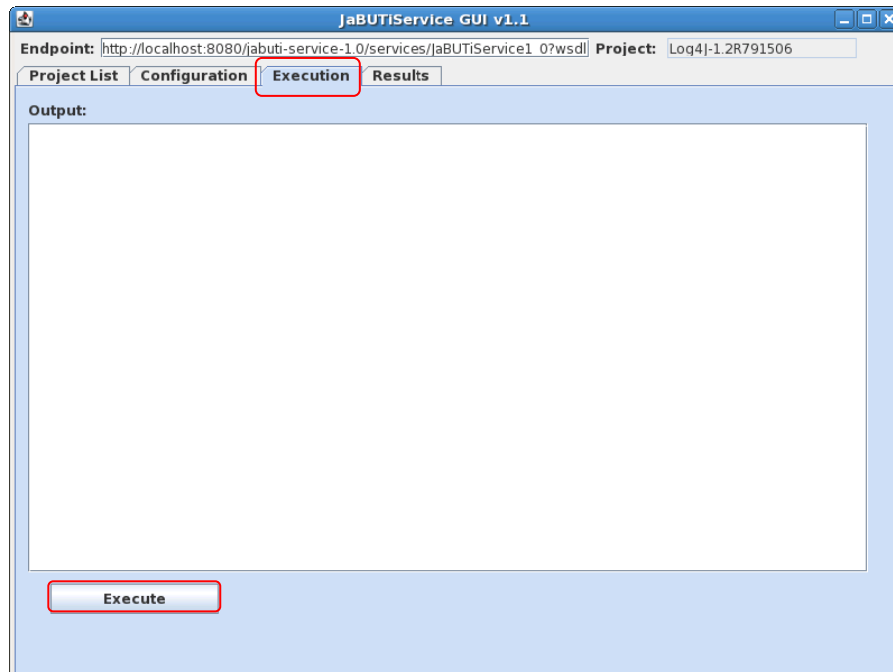


[Entry details] * Required entries only

- (1) Directory with original class: Path for directory under which test target classes are located
- (2) Directory of test script: Directory under which command specified in (4) is executed
- (3) Directory of test output: Output path for trace file (*.trc)
- (4) Command to execute the tests: Execution command of test scripts

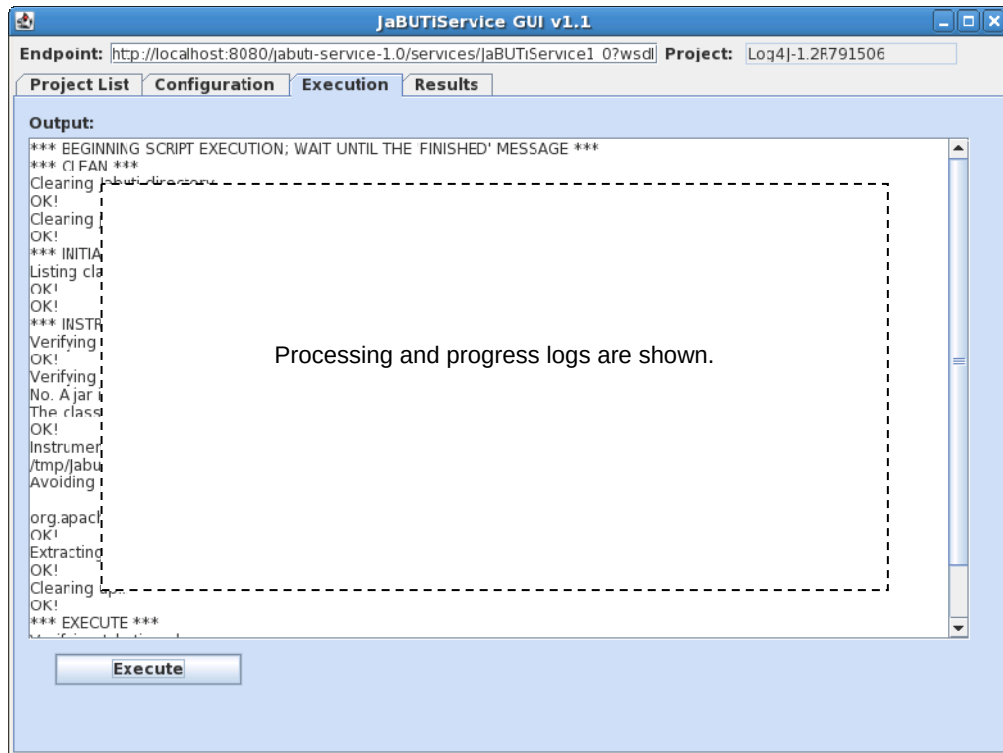
5. Select Execution tab, and click on Execute button.

Furthermore, if execution involves a project to which this automated test has once been performed, the project has to be rebuilt, as described in section "4.2.2 Preparing automated test (for JaBUTi only)".



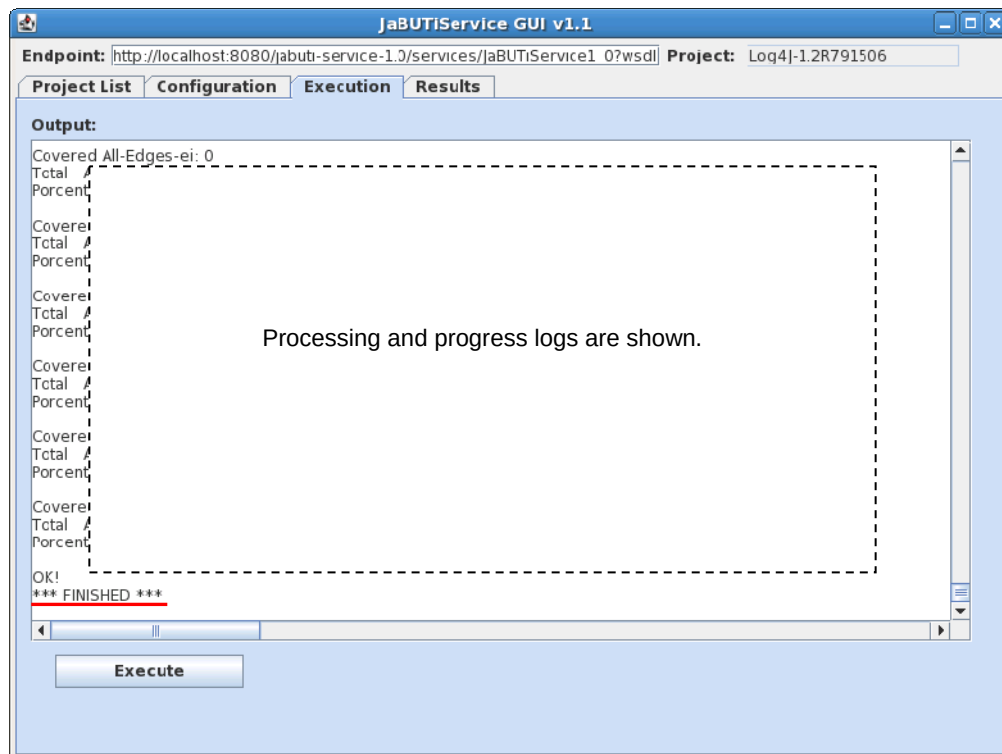
6. Measurement by JaBUTi starts.

Replacement to Class files to which coverage measurement processes are embedded, execution of automated test, and collection of coverage data are performed, along with log display.

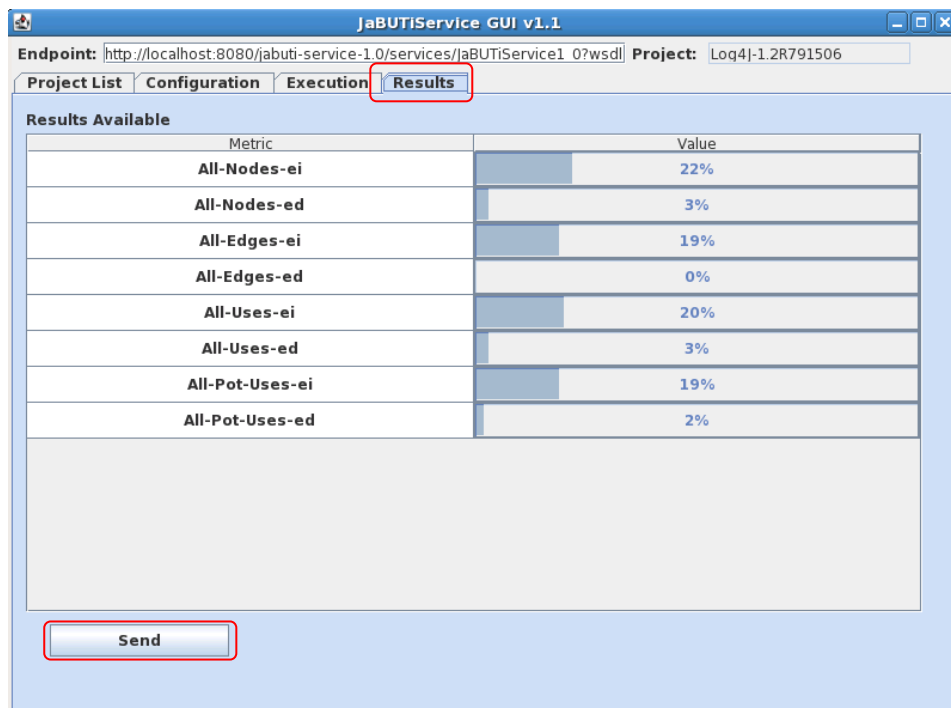


- Once the message "***FINISHED***" appears, all processes are completed.

Some projects may require several hours for this execution.



- Select Results tab, and click on Send button (which sends result to JaBUTi Web Service).

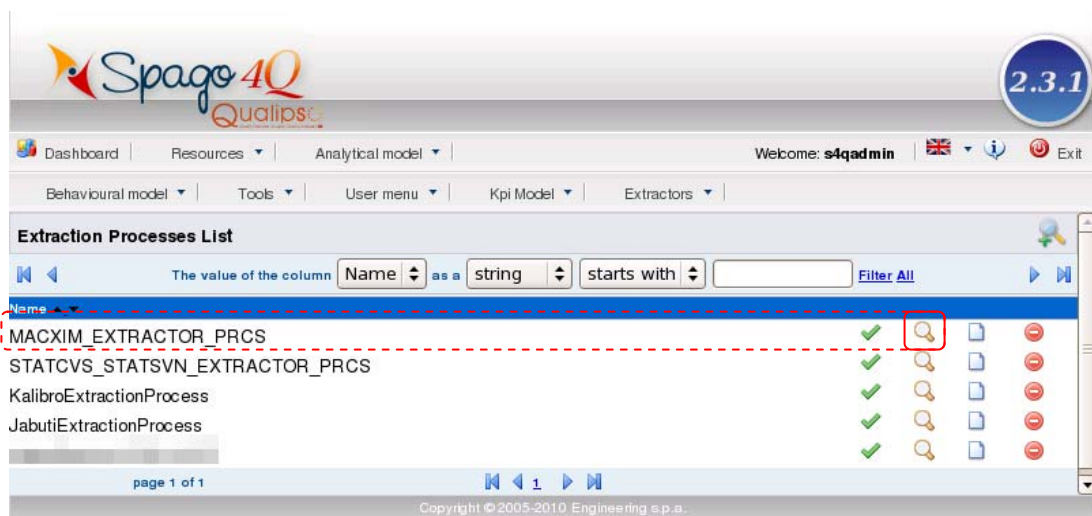


4.2.4. Analyzing project information

For analysis, run "Extractor" in the same manner as described in section "4.2.1 Extracting project information".

1. Log on to Spago4Q as administrator.
2. Select Spago4Q > Extractors > Extraction Processes and show list of evaluation tools.
3. Click on Select button  of tool you wish to use for evaluation.

Example) For evaluation using Macxim



For Name values which correspond to each tool, please refer to the following table:

Evaluation tool	Name
Macxim	MACXIM_EXTRACTOR_PRCs
StatCVS/StatSVN	STATCVS_STATSVN_EXTRACTOR_PRCs
Kalibro	KalibroExtractionProcess
JaBUTi	JaBUTiExtractionProcess

4. Extraction Process Detail screen appears. Click on Save and Execute button .
Example) For evaluation using Macxim

5. Once prompted that "processing may take time", confirm the message and click on "Yes".

6. Progress meter appears on the screen.

In some cases, internal processing may not be finished when progress meter completes.



Processing status can be confirmed on log file.

Example log file) [tomcat_root]/logs/catalina.out

[tomcat_root]/logs/SpagoBI.log

This operation allows analysis of measurement data produced by each tool using the process described in section "4.2.1 Extracting project information" and later. Once this execution completes, evaluation result can be verified.

4.3. Verifying evaluation result

Verify result of OSS project evaluation.

1. Show list of evaluation reports

Select Quality Platform > Actions > Quality Report, and show list of evaluation reports.

Select OSS's report you wish to verify from table.

Example) To view report of Checkstyle-5.0R2505 which has been evaluated using Macxim:



PROJECTS	REPORTS			
	StatSVN	Macxim	Kalibro	JaBUTi
APR-anonsvnRHEAD	StatSVN			
Ant-1.8.1R961732		Macxim		
Checkstyle-5.0R2505		Macxim		
CppUnit-1.12.1R582	StatSVN		Kalibro	
EvaluationDemo-0.5.5RHEAD				JaBUTi

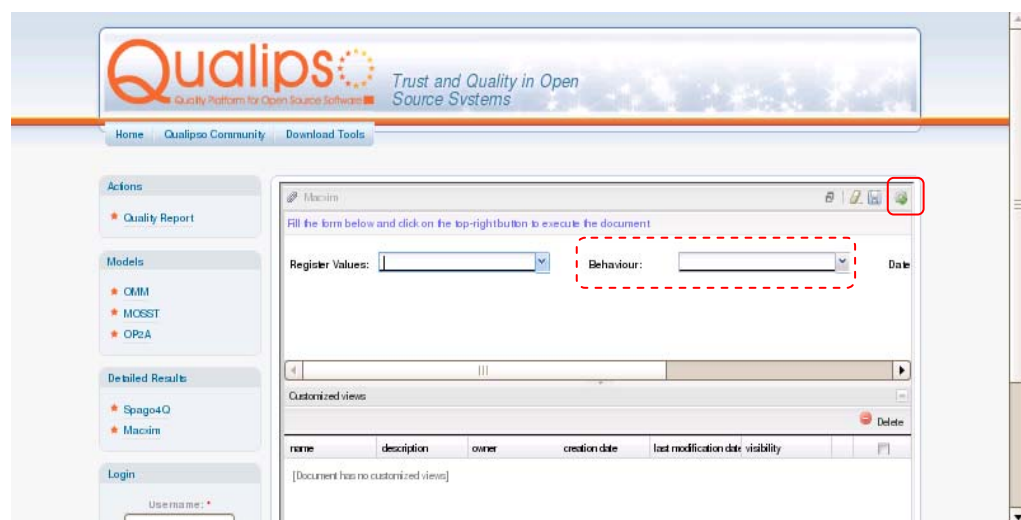
2. Verify evaluation result

Result display entry form appears.

Usually, leave "Behaviour" blank, and click on Execute Document  on upper-right.

If "Force Recalculation" is selected at "Behaviour", recalculation is performed and result is shown.

Please select this option when evaluation result is possibly different from the previous occasion where the result was shown.



Macxim

Fill the form below and click on the top-right button to execute the document

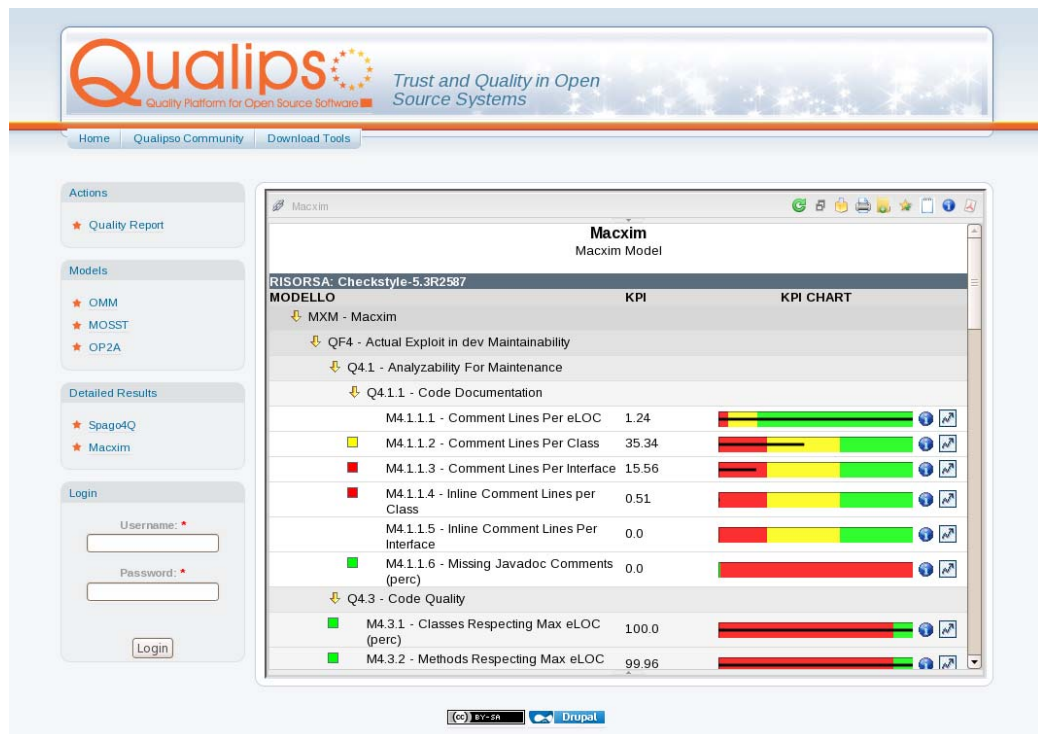
Register Values: Behaviour: Date:

Customized views

name	description	owner	creation date	last modification date	visibility
[Document has no customized views]					

(This page is blank)

Evaluation result appears.



* If evaluation result is not shown:

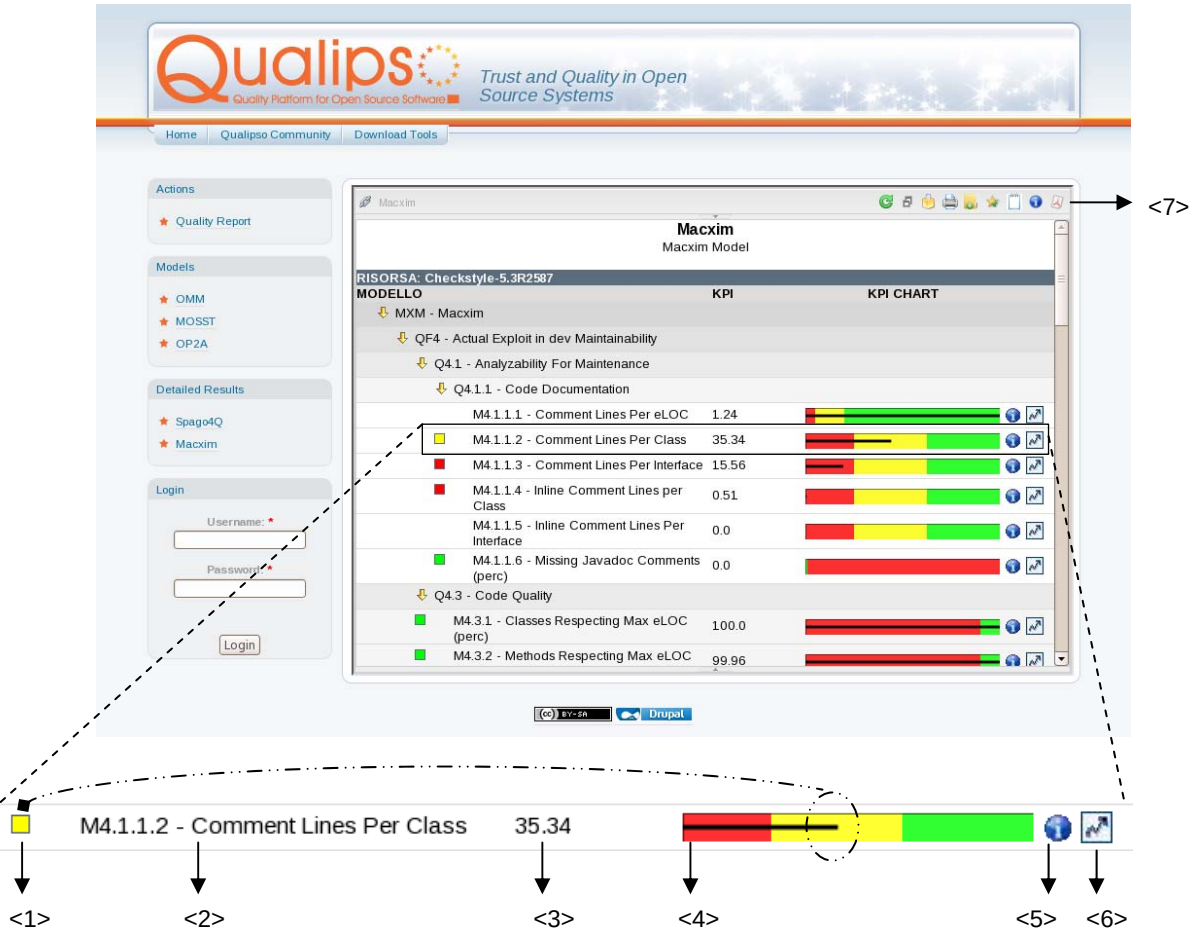
Processing described in section "4.2 Executing evaluation" may not be completed. Please retry after some time.

To show result again, please select "Force Recalculation" at "Behaviour".

5. Details of evaluation result

This section describes details of evaluation result.

Example) Details of report of Checkstyle-5.0R2505 which has been evaluated using Macxim:



[Description of each part]

<1> The color indicates which range in threshold area the current KPI value exists.

The color represents the position where black line of KPI chart described in <4> terminates.

<2> Model Instance name (evaluation scale)

In this example, "comment lines per class".

<3> KPI value

<4> KPI chart

- Threshold example

Red: Problem Yellow: Warning Green: Good

- Black line

Actual KPI value

<5> KPI overview



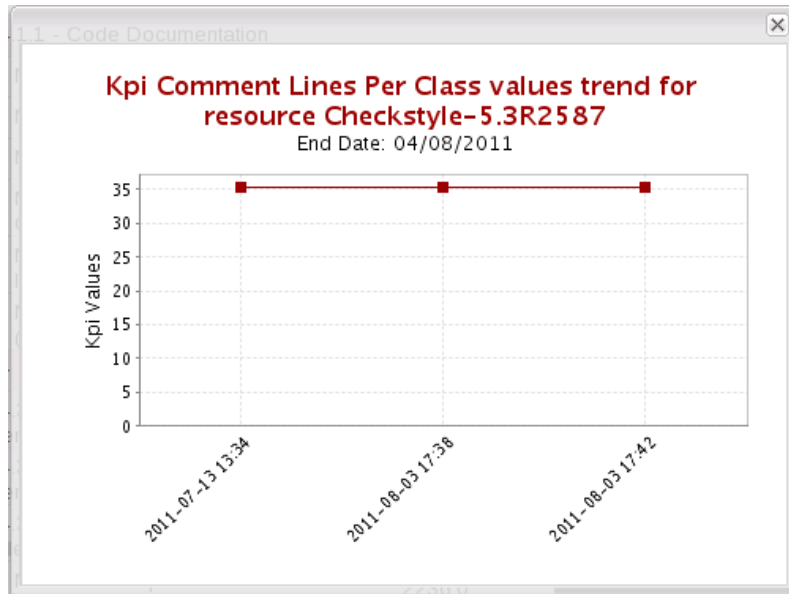
opens KPI overview window, in which threshold values described in <4> can also be verified.

KPI Overview	
Kpi Name	Comment Lines Per Class
Kpi Code	M_KPI_commentLinesPerClass
KPI Description	
KPI Interpretation	
Model Instance	
Technical Data	
Kpi Value Begin Date	2011-08-03 17:42:37.0
Kpi Value End Date	
Kpi	35.3302
Kpi Weight	
Kpi Weighted Value	
Kpi Value Description	
Kpi Value Target	
Threshold Name	commentLinesPerClass
Threshold RANGE Problem	0.0-20.0
Threshold RANGE Warning	20.0-50.0
Threshold RANGE Good	50.0-80.0

<6> Time line view



opens time line view window, in which temporal changes of KPI values can also be verified.



<7> Save as PDF



generates PDF file of evaluation result.

6. Changing project

This section involves operations performed by system administrator.

In general, MOSST tool does not allow to change/delete project once it is registered. However, if project name and version and revision are the same, other project information can be "overwritten", using the same procedure described in section "4.1 Registering project".

1. About changeable items

If project name, version, and revision are the same,
other project information (such as repository URL) can be overwritten.

Example) To change resource "Checkstyle-5.0R2505", which has project name "Checkstyle",
version "5.0", revision "2505", and repository type "svn", to the following registration:

- Project name	Checkstyle	
- Version	5.0	
- Revision	2505	
- Repository type	cvs	[Changed]

⇒ Resource name "Checkstyle-5.0R2505" remains the same, and repository type "cvs" is overwritten.

2. About unchangeable items

Once registered project name, version, and revision cannot be changed any more.

Please register a new project.

Example) To change resource "Checkstyle-5.0R2505", which has project name "Checkstyle",
version "5.0", revision "2505", and repository type "svn", to the following registration:

- Project name	Checkstyle	
- Version	5.0	
- Revision	2600	[Changed]
- Repository type	svn	

⇒ Resource name "Checkstyle-5.0R2600" is registered as new resource.

Registered information of project is available to verify by referencing database directly.

- Database: spago4q
- Table: QPS_PROJECT_DETAIL

7. Changing KPI definition

7.1. Changing KPI threshold value

MOSST allows to show graphical chart of OSS evaluation result by using KPI values, in color coding of red, yellow, green etc. (multiple level evaluation)

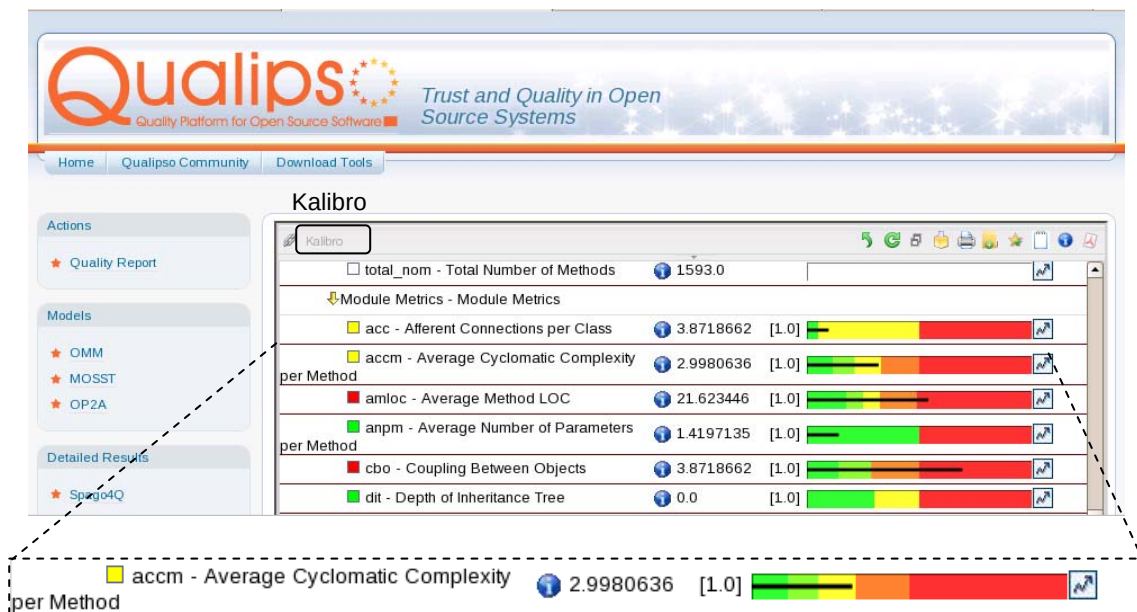
By default, sample threshold values provided by QualiPSo are set for evaluation.

Please change each threshold values in accordance with each evaluator's judgment criteria.

As an example, this section describes procedure to change range of "Warning" in evaluation result (5-level evaluation: Very good/Good/Regular/Warning/Bad) from 3.1-4.7 to 2.8-4.7, as well as to change display color of "Warning" from orange to violet.

1. Verifying current settings

- 1) On "5 Details of evaluation result" screen, click on ⓘ of KPI threshold value.



2) On pop-up screen, verify current KPI value and KPI threshold/display settings.

Technical Data	
Kpi Value Begin Date	2011-08-17 17:33:36.0
Kpi Value End Date	
Kpi	2.99806350668687
Kpi Weight	1.0
Kpi Weighted Value	2.9980636
Kpi Value Description	
Kpi Value Target	
Threshold Name	Kalibro ACCM Threshold
Threshold RANGE Very good	0.0-1.1
Threshold RANGE Good	1.1-2.0
Threshold RANGE Regular	2.0-3.1
Threshold RANGE Warning	3.1-4.7

3) In this example:

KPI value: 2.99806350668687

Regular range: 2.0-3.1 (yellow-colored)

Warning range: 3.1-4.7 (orange-colored)

can be observed.

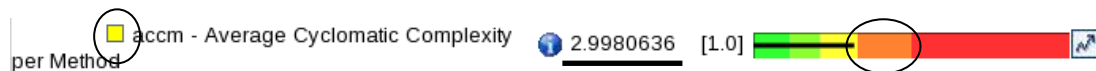
2. Change settings

Change boundary so that the current Regular KPI value "2.9980636" is made to Warning.

In addition, change display color of Warning in chart to violet.

[Current display]

5 levels: [VeryGood; Good; Regular; Warning; Bad]

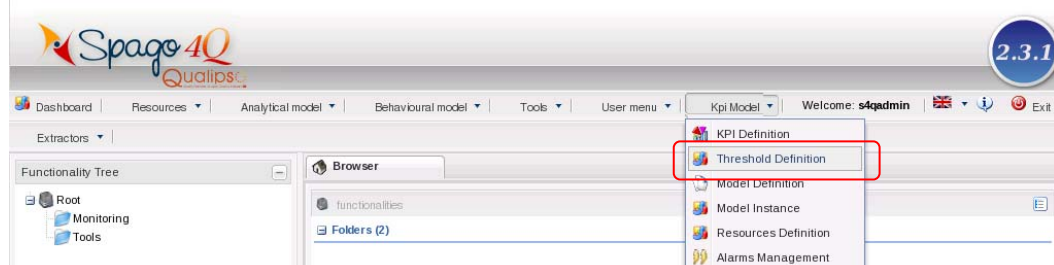


[Expected display after change]

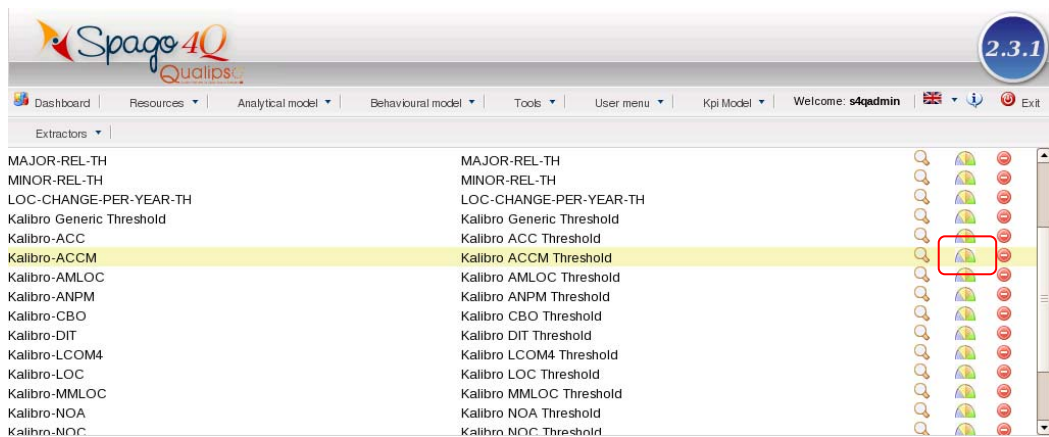


1) Log on to Spago4Q as administrator.

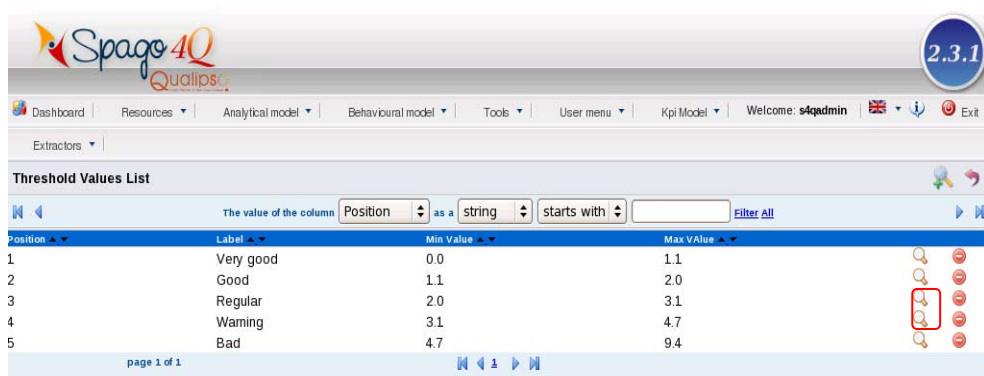
2) Select Spago4Q > KPI Model > Threshold Definition, and show Thresholds List.



3) From Thresholds List, select Code of KPI threshold value you wish to change, and then click on Threshold values button .



4) From Threshold Values List, select Position and click on Select button .



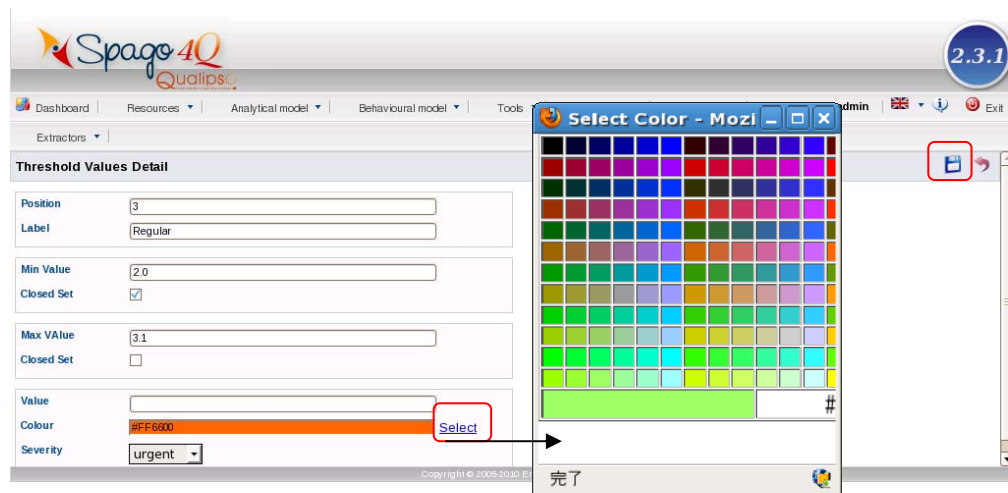
5) Settings are available for changes in Threshold Values Detail.

Once Select is clicked in Color section, Select Color screen appears, and color settings become available for change.

For Regular, set Min Value to 2.0 and Max Value to 2.8.

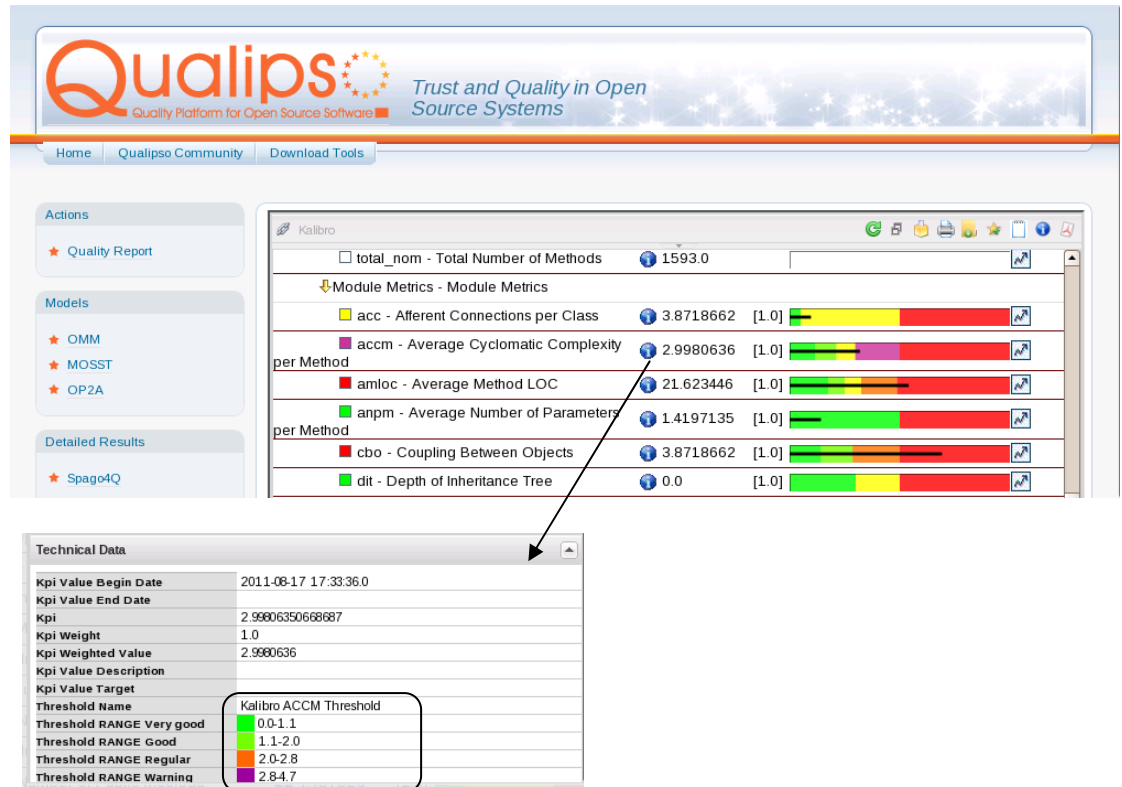
For Warning, set Min Value to 2.8 and Max Value to 4.7.

Set Color to violet, and then click on Save button .



(This page is blank)

- 6) Show evaluation result again in accordance with steps described in "7.1 Verifying current settings" of this section, and verify changes.



8. Frequently asked questions about troubles (FAQ)

8.1. When executing Extractor on Spaga4Q

1. I run Extractor for StatCVS on Spago4Q (STATCVS_STATSVN_EXTRACTOR_PRCs), but evaluation report is not shown. "IllegalDataException" is logged in catalina.out.

A byte string "0x1B" may be included in commit log which OSS repository returns after execution of Extractor for StatCVS. "IllegalDataException" occurs because "0x1B" is not supported by jdom library which is bundled with StatCVS.

An example of "0x1B" included in commit log is OSS project which uses JIS code set (Japanese) for commit log (JIS code set uses a byte string which includes "0x1B" as a code to switch between 1-byte characters and 2-byte characters.)

As of August 2011, StatCVS used in MOSST does not support this case.

Information

When this case is applicable, the following exception is logged in catalina.out.

Example: After Extractor was run for StatCVS, commit log was successfully retrieved from namazu's CVS repository (cvs -Q log), but Java exception occurred while returned commit log was being analyzed:

```
20:29:28,168 DEBUG ProcessWrapper:42 - Starting process: cvs -Q log
20:30:00,039 DEBUG PSNCStatGenerator:55 - Temp dir for statTool:
/usr/local/tomcat/temp/statDir381914070418216938
20:30:00,039 INFO AbstractLogTool:36 - Executing stat task, log:
/usr/local/tomcat/temp/log4002628011430765802, projectDir:
/tmp/StatToolsService/RepoHolder/namazu/stable-2-0/namazu, out:
/usr/local/tomcat/temp/statDir381914070418216938
2011/07/29 20:30:00 net.sf.statcvs.Main generateDefaultHTMLSuite
¥u8b66¥u544a: Log and working copy are out of sync. Reports will be inaccurate.
20:30:00,590 ERROR StatsToolsExtractor:138 - Generate statistics error
psnc.se.extraction.err.ExtractionException: org.jdom.IllegalDataException: The data "Add *
[$B%$%s%G%C%/%9:n@.$N8zN(2=^[([B [namazu-dev 1088]" is not legal for a JDOM CDATA section:
0x1b is not a legal XML character.

at psnc.se.extraction.PSNCStatGenerator.generateStatistics(PSNCStatGenerator.java:62)
at psnc.se.extraction.StatsToolsExtractor.analyzeOne(StatsToolsExtractor.java:131)
at psnc.se.extraction.StatsToolsExtractor.extractOne(StatsToolsExtractor.java:103)
```

```
at psnc.se.extraction.ws.ExtractionManager.analysisRun(ExtractionManager.java:155)
at psnc.se.extraction.ws.ExtractionManager$AnalysisTask.call(ExtractionManager.java:189)
at psnc.se.extraction.ws.ExtractionManager$AnalysisTask.call(ExtractionManager.java:176)
at java.util.concurrent.FutureTask$Sync.innerRun(FutureTask.java:303)
at java.util.concurrent.FutureTask.run(FutureTask.java:138)
at java.util.concurrent.ThreadPoolExecutor$Worker.runTask(ThreadPoolExecutor.java:886)
at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:908)
at java.lang.Thread.run(Thread.java:662)
```

2. I run Extractor of evaluation tool on Spago4Q, but evaluation report is not shown. No log is generated in catalina.out for checkout operation.

If your network environment requires Internet access via proxy server, source code checkout may have failed. Please verify if there is any error in proxy server settings.

- (1) If repository type is svn, please confirm that proxy server setting is correctly made for svn command, with the following procedure.
 - (a) Set proxy server to /root/.subversion/servers file for svn command.
http-proxy-host = [Proxy server host name]
http-proxy-port = [Proxy server port number]
 - (b) Restart Tomcat.
- (2) If the problem still exists, please confirm that proxy server setting is made correctly for JavaVM, with the following procedure.
 - (a) Set proxy server to [tomcat_root]/bin/setenv.sh file for Tomcat (JavaVM).
To CATALINA_OPTS, add "-DproxySet=true -DproxyHost=[Proxy server host name]
-DproxyPort=[Proxy server port number] -Dhttp.nonProxyHosts=localhost".
 - (b) Restart Tomcat.
- (3) If the problem still exists, please confirm that proxy server setting is made correctly, with the following procedure.
 - (a) Set proxy server to /etc/profile file.
http_proxy= [Proxy server URL]:[Proxy server port number]
https_proxy= [Proxy server URL]:[Proxy server port number]
 - (b) Restart Tomcat.
- (4) If the problem still exists, please manually run "svn" or "cvs" command directly, to see if source code can be checked out (downloaded).
Example of svn) # svn co http://svn.ruby-lang.org/repos/ruby/trunk ruby
Example of cvs) # cvs -d:pserver:cvs@pserver.samba.org:/cvsroot co -r SAMBA_2_2 samba

If "Unknown host pserver.samba.org." or other error code such as the following (405 etc.) is returned, the problem may reside in the network environment or repository you are connected to. Please contact network or repository administrator for more information.

"svn:server sent unexpected return value (405 invalid method) in response to PROPFIND request for '/repos/ruby/trunk'"

Information

Please also refer to "QualiPSo MOSST Installation Manual".

8.2. When showing report of evaluation result

1. Report of evaluation result from the tool does not appear on Quality Platform (or, report is not updated to the latest data).

When evaluation result is possibly different from the previous occasion where result was shown, recalculation of evaluation report is required. Please verify if the problem can be avoided by the following procedure:

- (1) Select Quality Platform > Actions > Quality Report.

Click on report you wish to show.

- (2) Select "Force Recalculation".

In "Behaviour" pull-down, select "Force Recalculation" and click on Execute Document on upper-right.

Information

Please also refer to "4.3 Verifying evaluation result" in this document.

8.3. Others

1. "OutOfMemoryError" is logged in catalina.out.

Memory allocated to Tomcat (JavaVM) may be insufficient. Installation Manual uses recommended memory size, but it is possible to change it in accordance with user environment. Please increase memory allocation size and verify if the problem can be avoided.

(1) [tomcat_root]/bin/setenv.sh

- At setting in CATALINA_OPTS="-Xms1024m -Xmx1024m -XX:MaxPermSize=256m", please increase memory allocation size.
- Size of heap memory area can be set by -Xms and -Xmx.
- Size of permanent memory area can be set by -XX:MaxPermSize.

Information

Please also refer to "QualiPSo MOSST Installation Manual".

9. References

- MOSST
 - <http://www.qualipso.org/mosst-champion>
- Spago4Q
 - <http://www.qualipso.org/spago4Q-tool>
 - <http://www.spago4q.org>
 - <http://wiki.spago4q.org/xwiki>
- Quality Platform
 - <http://www.qualipso.org/node/538>
- Kalibro
 - <http://www.qualipso.org/kalibro-tool>
 - <http://ccsl.ime.usp.br/kalibro>
- StatCVS/StatSVN
 - <http://www.qualipso.org/stat-tools>
 - <http://www.statsvn.org/>
 - <http://statcvs.sourceforge.net/>
- Macxim
 - <http://www.qualipso.org/macxim-tool>
 - <http://qualipso.dscpi.uninsubria.it/macxim>
 - <http://macxim.qualipso.org/>
- JaBUTi
 - <http://www.qualipso.org/JaBUTi-tool>